用户行为分析

判定用户是否在某一商家就餐过的标准是，该用户是否对商家进行过评论。有过评论即代表在此餐厅就餐过。1条评论代表1次就餐记录。

**用户就餐地理位置迁移网络**

**数据：business.json（保留business-id ,name ,city, latitude, longitude, star, categories**

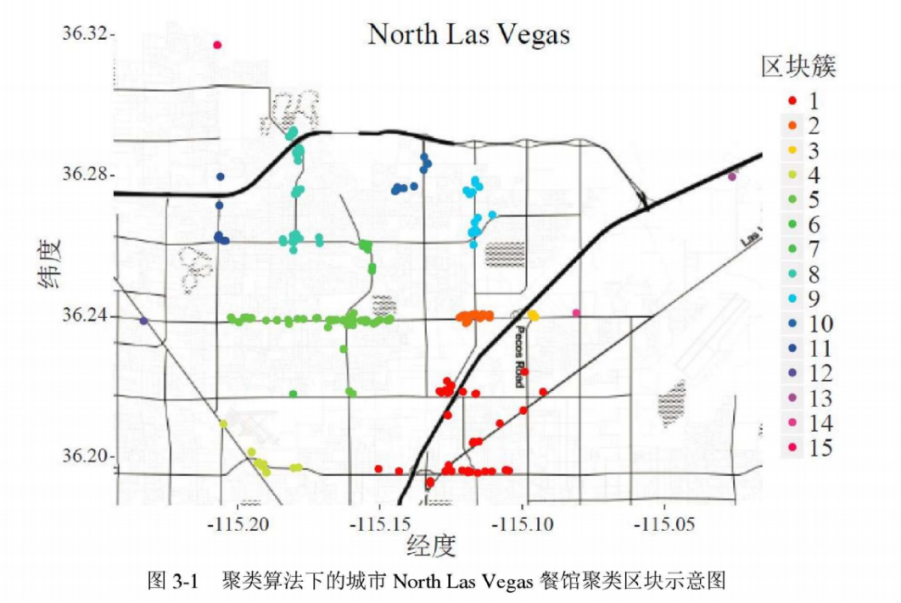
**商家ID、名称，商家所在城市、经纬度，商家评分、口味标签，）**

**review.json（保留review-id, user-id, business-id, star**

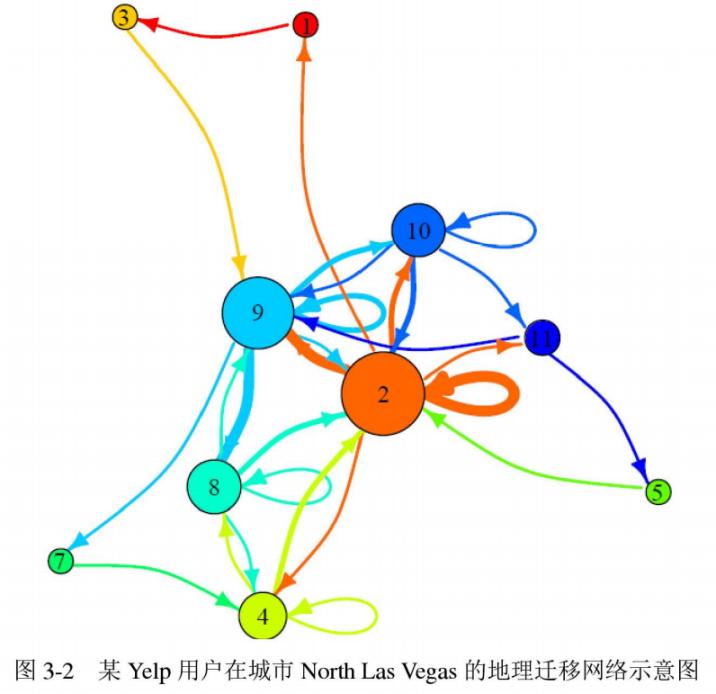
**评论ID，用户ID，商家ID，用户对商家评分即评论评分）**

**用户数据已进行预处理，筛选出Boston、Columbus两个城市的商家，以及与这两个城市所有商家相关的评论数据**

1. 对boston及columbus两个城市商家分别进行聚类（使用基于密度聚类-DBSCAN <https://github.com/guoziqingbupt/DBSCAN> ），输出聚类结果（数据即可）。
2. 选取具体某一个城市，例如Boston，进行该城市的餐厅聚类展示，输出图例。例如下边这样，或者有其他的展示形式也可以。



1. 选取一个在Boston有就餐记录的用户（即在review.json中对Boston餐厅进行过评论的用户），构建该用户在Boston所有就餐记录就餐地理位置迁移网络。输出迁移图，例如：



**用户就餐口味迁移网络**

1. 选定一个用户
2. 提取其到过的**所有**餐厅{R1,R2,……,Rn}
3. 针对商家口味标签进行用户每次就餐偏好标签提取（高斯核密度估计）
4. 构建口味迁移网络

高斯核密度估计：

|  |
| --- |
| 用户每次就餐偏好口味提取 |
| 输入：Yelp用户u的完整用餐历史记录{,,…,}(## 用户就餐的T个餐厅）,包括每个餐馆i口味标签（## 代表餐厅的个口味标签）及评分（## 是用户u给餐厅i的评分，我们把这个分数默认为餐厅i的K个口味标签得分都是）。令为该用户的就餐口味集合（## 用户U去过的T个餐厅的所有口味标签）。  输出：该用户每次用餐的最中意的口味序列{。  1：假设该用户每次用餐的评分对该餐馆的每个标签均有效，根据各个口味的历史评分记录，计算每个口味的评分高斯分布N（）（## 通过口味标签的得分来计算期望和方差）  2: for i = 1:T do （对商家）  3: for j = 1: do （对口味标签）  4: 对当前口味且,计算/  5: end for  6: if min{|  7: = （# ，都是数值，是标签）  8: end if  9: end for  10: return {} |

然后再分别对两个城市所有用户进行上述两个网络构建，最终融合城市所有用户的网络得到该城市用户的就餐地理位置迁移网络和用户就餐口味迁移网络。

网络表征学习

利用deepwalk算法，分别对用户就餐地理位置迁移网络和口味迁移网络进行网络节点的表征学习，得到每个地理区块所对应的嵌入向量，以及每个口味标签所对应的嵌入向量

融合用户行为信息进行推荐

将每个餐馆的地理-口味标签（,）分别用嵌入向量和进行表征

因此用户就餐的餐厅序列{R1,R2,…,Rn}可以表示为：



然后计算用户u在位置和口味向量空间上的就餐喜好中心，公式为：



接着计算用户每次就餐行为的怀旧值

（相当于传统协同过滤算法中用户对物品的打分，在这个方法里，就是相当于用户对餐厅的打分）

基于用户：预测用户u对未评分物品i的评分值

其中，S（u,K）表示与用户u兴趣相近的K个近邻用户，N（i）表示对物品i有过行为的用户集合，wuv表示用户u与用户v的相似度，rvi表示用户v对物品i的评分，在本研究中即为用户对餐厅的评分。

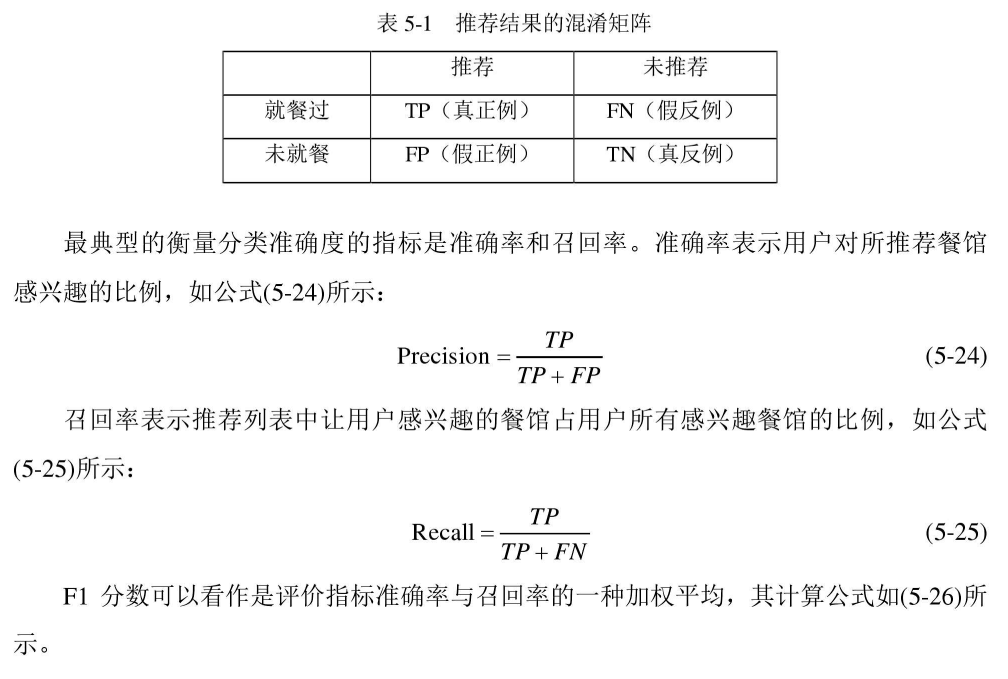
基于物品：计算用户u对餐馆i的评分预测值

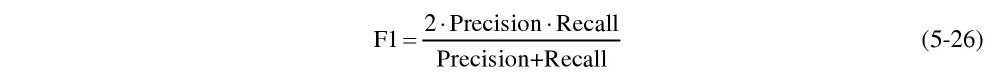
其中，N（u）表示用户u评价过的餐馆集合，S（i,K）表示餐馆i最相关的K个餐馆集合，wij表示餐馆i与餐馆j的相似度，rui表示用户u对餐馆i的评分值。

本文是对传统协同过滤算法的改进，融入用户行为信息的（基于用户的协同过滤/基于物品的协同过滤）其实就是把求出来的这个当成传统协同过滤算法中的评分，这样说可能比较好理解

最后的推荐部分使用的对比推荐算法分别为：传统的基于用户的协同过滤和基于物品的协同过滤。

评价指标





需要输出的内容：

1. 具体一个城市的餐厅聚类展示
2. 具体某一用户的就餐网络迁移、就餐口味迁移
3. 最终的推荐结果（其实从城市所有用户网络融合开始就已经是在为最后的推荐进行前期处理工作了）